UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/972,362	10/05/2001	Farhan Ahmad	SJ09-2001-0099	4699

46917 7590 05/23/2008

KONRAD RAYNES & VICTOR, LLP.
ATTN: IBM37
315 SOUTH BEVERLY DRIVE, SUITE 210
BEVERLY HILLS, CA 90212

EXAMINER	
DIVECHA, KAMAL B	
ART UNIT	**PAPER NUMBER**
2151	

MAIL DATE	DELIVERY MODE
05/23/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 09/972,362
Filing Date: October 05, 2001
Appellant(s): AHMAD ET AL.

David W. Victor
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 3/14/08 appealing from the Office action mailed 9/4/07.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

Weber et al., US 6,480,901, filed on Jul. 09, 1999 and issued on Nov. 12, 2000.

Ismael et al., US 6,851,118 B1, filed on Jun. 27, 2000 and issued on Feb. 01, 2005.

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claims 1-2, 4-9, 21-48 are rejected under 35 U.S.C. 103(a) as being unpatentable over Weber et al. (hereinafter Weber, U. S. Patent No. 6,480,901) in view of Ismael et al. (hereinafter Ismael, US 6,851,118 B1).

As per claim 1, Weber discloses a system in communication with a network comprising one or more network components (i.e. storage devices and one or more hosts via a switching fabric component, fig. 1, col. 1 L25-58, col. 6 L45-54), comprising:

a manager in communication with the network components (fig. 1 and col. 2 L3-8, col. 2 L22-44); and

an interface process (such as management and/or user interface) in communication with the manager and the network components (a switching fabric component and the hosts) wherein the interface process performs (col. 13 L1-9, fig. 5 item #510, col. 2 L22-44, col. 9 L31-35):

obtaining information on the network components from the manager (col. 13 L9-49, col. 4 L34-51, col. 7 L15-65, col. 9 L59-66);

maintaining in a storage device information identifying application process residing on the network components and communication interface types and a parameter name, wherein the parameter name is used with the communication interface type to

invoke the application process residing on the network component (col. 6 L55-67, col. 9 L10-42, col. 16 L51 to col. 17 L37: using the specified management interface version for launching the control software, fig. 2 item #208, col. 13 L1-49, col. 15 L33 to col. 16 L67);

displaying information representing the network components (col. 13 L9-49, col. 4 L34-54, col. 7 L15-30, col. 15 L14 to col. 16 L50, fig. 6: shows the graphical view);

receiving selection of one displayed network component (col. 16 L51 to col. 17 L34, fig. 6: double clicking or selecting the devices in management domain window displays the detailed information window, col. 16 L51-67);

accessing the storage to determine at least one application process associated with the selected network components (col. 16 L51 to col. 17 L51: the DMA receives device property information about the selected component from storage area);

displaying information on the at least one determined application processes associated with the selected network component, wherein at least one of the determined application processes reside on the selected network component (col. 13 L1-49, col. 16 L58-67: the DMA displays in the device properties the storage system's management interface program version required to execute the control software of network component, i.e. the management application program associated with the device);

receiving selection of one the displayed application processes (col. 7 L25-39, col. 13 L1-49 and col. 16 L51 to col. 17 L35: i.e. receiving selection of one of the displayed application processes information);

accessing the applet repository to determine information on the selected application process and the communication interface type and parameter name supported by the application process, i.e. the applet to use to launch the selected application process on the selected network component (col. 13 L64 to col. 14 L41, col.16 L51 to col. 17 L35: accessing the storage to determine the applet/management interface program to use to launch the application/control process using the specified version); and

launching selected application process on the selected network component using the determined communication interface type and parameter name from the storage (col. 13 L1-49 and col. 7 L25-39, col. 16 L51 to col. 17 L35: using the specified version/name from storage, col. 26 L13 to col. 27 L19).

However, Weber does not disclose a rules file having at least one rule for each of the network components, wherein each rule identifies the network component to be managed, one of a plurality of communication interface types and a parameter name, wherein the parameter name is used with the communication interface type to invoke the application process residing on the network component (i.e. Weber does not expressly defined the process wherein the rules file identifies application processes residing on the network components. Note that Weber does disclose identifying and executing the application process specific to the managed device, from the storage, logically, this implies that there must be a file in the storage. According to applicant specification, this rule file is an xml file, see the disclosure above).

Ismael discloses maintaining a rules file in a storage device having at least one rule for each of the network components or device, wherein each rule identifies the network component to be managed, one of a plurality of communication interface types and a parameter name,

wherein the parameter name is used to with the communication interface type to invoke the application process or program residing on the network components, i.e. the remote invocation of the program using the objects (col. 7 L44 to col. 8 L67: e.g. html adaptor in form of html file is initiated at the remote device through the html object adaptor, wherein the object identifies the communication interface and parameter name used to invoke the adaptor, col. 9 L30-59, col. 18 L20-60: initializing an adaptor involves using the parameter such as port number, logical name, etc., See also, col. 1 L50 to col. 2 L41).

Therefore, it would have been obvious to a person of ordinary skilled in the art at the time the invention was made to modify Weber in view of Ismael in order to identify a network component to be managed, one of plurality of communication interface type and a parameter name, wherein the parameter name is used with the communication interface type to invoke the application on the network component.

One of ordinary skilled in the art would have been motivated because it would have provided a system of accessing from a client station a target object at a remote station via network in order to manage the remote station(s) (Ismael, col. 1 L39-55, col. 2 L41-54) and/or it would have provided a computer-implemented method of accessing from a client machine an object at a remote machine using a browser (Ismael: col. 9 L3-30).

As per claim 2, Weber discloses a graphical output device coupled to the interface process for displaying one or more graphical objects representing the application processes on the network components, wherein the interface process is coupled to the graphical output device for effecting the display of the graphical objects on the graphical output device (col. 13 L1-67 and col. 4 L34-51, fig. 6, col. 9 L10-42, col. 16 L51 to col. 17 L35).

As per claim 4, Weber discloses the system wherein the interface process responds to selection of one of the objects representing one application process by effecting execution of the application process represented by that object (col. 13 L1-49 and col. 14 L10-15).

As per claim 5, Weber discloses a store containing information regarding one or more network components and one or more application processes residing on the network component (col. 2 L26-36, col. 8 L27-47, col. 16 L51 to col. 17 L35: applet repository stores management applications programs).

As per claim 6, Weber discloses the system wherein the interface process accesses the store, upon selection of one graphical object representing one of the network component, to identify one application process residing on the selected component represented by the selected object (col. 13 L1 to col. 14 L15 in conjunction with fig. 6).

As per claim 7, Weber discloses the system wherein the application process is any of an executable application, a web-browser application, a telnet session, or an SNMP application (col. 10 L54-65, col. 6 L15-26, fig. 4 item #406, fig. 5 item #510, 512).

As per claim 8, Weber discloses the system wherein the information on the network components includes an identifier for the network component and application processes residing on the network component (col. 13 L10-49, col. 16 L51 to col. 17 L35, fig. 6).

As per claim 9, Weber discloses the system wherein at least one of the graphical objects representing one network component provides a textual description of that network component (fig. 6 and col. 13 L37-67, col. 14 L23-40).

As per claim 38, Weber discloses the system wherein the information in the storage for at least one network component is obtained from an operator administrator and the information in

the storage for at least one other network component is obtained via standardized queries of the at least one other network component (col. 7 L15-59, col. 9 L10-67 and col. 15 L14 to col. 16 L19).

As per claim 39, Weber discloses the system comprising displaying information on the at least one determined application process on the network component to enable selection of the application process on the selected network component to launch (col. 13 L1-49 and col. 7 L25-39, col. 16 L51 to col. 17 L35, col. 26 L13 to col. 27 L19).

However, Weber does not disclose the system comprising displaying information on a plurality of application processes residing on the selected component (i.e. if the device is associated with multiple application programs).

But it would have been obvious to a person of ordinary skilled in the art at the time the invention was made to modify Weber in order to display plurality of application processes residing on the selected network component if the network component is associated with more than one application process, because Weber teaches the process of displaying information on application process residing on the selected network component.

One of ordinary skilled in the art would have been motivated because it would have enabled the network administrator to launch one of the application process residing on the network component in order to manage the device (Weber, col. 16 L51 to col. 16 L16).

As per claim 40, Weber discloses the system wherein the network components comprise hosts, storage devices, and at least one switching fabric, wherein the manager communicates with the hosts and storage devices via the at least one switching fabric (fig. 1, col. 1 L25-58, col. 6 L45-54).

As per claim 41, Weber in view of Ismael discloses the system wherein the parameter name comprises an address used to communicate with the network component to invoke the application process (Ismael: col. 18 L20-65).

As per claim 42, Weber in view of Ismael discloses the system wherein the parameter name comprises an executable name of the application process residing on the network component (Ismael: col. 18 L20-65).

As per claims 21-37 and 43-48, they do not teach or further define over the limitations in claims 1-2, 4-9 and 38-42. Therefore claims 21-37 and 43-48 are rejected for the same reasons as set forth in claims 1-2, 4-9 and 38-42.

(10) Response to Argument

The examiner summarizes various arguments raised by the appellant and addresses replies individually.

In the Brief, appellant argues in substance that:

a. Weber does not teach receiving selection of a displayed application process residing on a selected network component (Brief, pg. 21 [VII. A.]).

In response to argument [a], Examiner respectfully disagrees.

Independent claim 1 recites:

A system in communication with a network comprising one or more network components comprising:
a manager in communication with the network components having application processes residing on the network components; and
an interface process in communication with the manager and the network components, wherein the interface process performs:
obtaining information on the network components from the manager;
maintaining a rules file having at least one rule for each of the network components, wherein each rule identifies the network component to be managed, one of a plurality of communication interface types, and a parameter name, wherein the parameter name is used with the communication interface type to invoke the application process residing on the network component;
displaying **information representing** the network components;
receiving selection of one displayed network component;
accessing the rules file to determine at least one application process associated with the selected network component;
displaying information on the at least one determined application process residing on the selected network component, wherein at least one of the determined application processes reside on the selected network component;
receiving selection of one of the displayed application processes residing on the selected network component;
accessing the rule from the rules file for the selected application process to determine information on the selected application process and the communication interface type and parameter name supported by the application process to use to launch the selected application process on the selected network component; and
launching the selected application process on the selected network component using the determined communication interface type and parameter name from the rules file.

For support, appellant's disclosure is reproduced herein: (specification, pg. 174-177):

Launching Device Specific Applications

As discussed above, a SAN according to the invention can include a variety of components, such as one or more digital data processors hosts, one or more storage device, and a switching fabric, having a variety of components, such as, switches, hubs, gateways, for providing communication between the hosts and the storage devices. **These components are typically acquired from different vendors, and have various application software associated therewith**. For example, the switching fabric components can have vendor-specific management applications that allow configuring and/or managing these components.

The illustrated embodiment permits the SAN operator/administrator to execute these vendor-specific applications from a single graphical user interface, to wit, that SAN manager GUI 20, in a manner described in more detail below.

With reference to FIGURE 6 and FIGURE 42, the SAN manager service 38 maintains a representation of the SAN that provides information, inter alia, regarding the identity of the SAN components, and the connectivity of these components. In addition, the manager service 38 maintains for selected components, for example, the switching fabric components, information regarding management applications specific to them. **These can be applications, by way of non- limiting example, residing directly on the components, applications invoked or effected through HTTP, telnet or other servers residing on the components or on proxy services residing elsewhere**, and/or via applications running on the SAN manager itself. **This information is stored, for example, in a file, referred to herein as a "Rules" file, which identifies each of the selected components and the applications and communication interfaces supported by that component, e.g., telnet, SNMP.** In the illustrated embodiment, **a mark-up language, e.g., XML**, is utilized to format the information contained in the Rules file, though in other formats may be used instead or in addition.

Information regarding the component management applications can be obtained from the operator/administrator (e.g., via prompt and/or menu option when the respective components are first added to the system or subsequently) and/or obtained directly from the components themselves. In the case of the latter, the information can be obtained via standardized queries, such as Management Server queries or FC MANAGEMENT MIB queries. ln the case of components that cannot respond to such queries with the necessary information (as where the corresponding management application resides on the SAN manager itself) and/or that have multiple management applications, any information obtained from the component is augmented in the Rules file with information, e.g., obtained from the operator/administrator, identifying the necessary or preferred application.

The Netview server can effect retrieval of the SAN representation from the manager service 38 and the display of selected information discerned from the retrieved representation on the Netview console 52, as described in detail above. In one embodiment, the Netview console 52 displays a plurality of graphical objects, e.g., icons, each of which represents one of the SAN components. Alternatively, a textual list of the SAN components can be displayed. Further, the Netview console 52 provides an operator, e.g., the SAN administrator, with a user interface) element, such as keyboard or mouse, that permits selection of one of the displayed components.

The Netview server allows the operator to launch an application process associated with a selected SAN component, such as, a management application residing on that component, such as, a switch, in a manner described below. In response to the selection of a graphical object representing a SAN component, the Netview server accesses the Rules file to obtain information regarding the application processes associated with that selected component, and effects the display of this information, for example, in the form of a menu, on the Netview console 52. ln some embodiment, a plurality of management applications residing on a selected component are displayed while in other embodiments, only the primary management application is displayed. To facilitate the display of information regarding on the SAN components on the Netview console, in some embodiments, **the Netview server stores the information retrieved from the SAN manager service 58 regarding the applications residing on the SAN components in a persistable storage.**

The Netview server 54 responds to the selection of one of the displayed application processes by effecting the launching of that application process via an interface process, such as a web-based browser application, a telnet process, or an SNMP application. More particularly, the Netview server 54 communicates with the SAN manager service 38 to retrieve information, such as, launch method and its respective parameters, therefrom. **The SAN manager service responds to a request from the Netview server for the launch**

information by parsing the Rules file to generate an object, e.g**., an XML object** that contains the requisite information, and transmits the information to the Netview server. The Netview server utilizes the object returned from the SAN manager service to effect the launching of the selected application process. **Once the selected application, e.g., a management application, is launched**, the operator can utilize the application, via the interface software provided by the Netview server, to configure and/or manage the SAN component on which the application resides. This advantageously allows the operator, e.g., the SAN administrator, to manage a variety of SAN components, having different management applications, from a single entry point, that is, from the Netview server/console.

Claim Interpretation

In interpreting the independent claim 1, the followings should be noted:

> "…displaying **information representing** the network components;
> receiving selection of one displayed network component;
> accessing the rules file to determine at least one application process associated with the selected network component;
> **displaying information on the at least one determined application process** residing on the selected network component, wherein at least one of the determined application processes reside on the selected network component;
> **receiving selection of one of the displayed** application processes residing on the selected network component;
> accessing the rule from the rules file for the selected application process to determine information on the selected application process…"

First, the recited limitation in the claim refers to "displaying information on the at least one determined application process residing on the network component..."

Secondly, "…receiving selection of one the displayed…" is with respect to the information displayed.

In short, the claim calls for **displaying information** on the at least one determined application process on the network component and **receiving selection of one of the displayed information on** the one determined application process.

Weber et al.

BACKGROUND OF THE INVENTION

The present invention relates generally to methods and apparatus for managing devices on a network, and more particularly to a network based system and software for monitoring and managing devices that are not attached directly to the network, but are proxy attached to the network via another device.

Network computing systems typically require a variety of devices to construct and maintain a working storage system. In addition, companies with large networks typically have a number of different storage systems, many of which can be manufactured by different companies and/or run on different versions of operating software. Storage system devices may include, but are not limited to, host adapters, I/O chips, disk enclosures, and bridge controllers, to name a few.

Each of these components traditionally are managed by proprietary software that is supplied by its manufacturer. In addition, there are a number of third parties which have developed network management frameworks, such as Hewlett-Packard's Open View, IBM's NetFinity, and Computer Associates' Unicenter. Unfortunately, however, while these third party frameworks provide great benefit to the management of applications, servers, and network equipment, they have little success in managing storage devices because no single standard exists for configuring and monitoring storage devices produced by different manufacturers, as well as different versions of storage devices produced by the same manufacturer. Standards such as desk top management interface (DMI) and simple network management protocol (SNMP) are able to manage simple devices such as host adapters and the like, but they fall short when applied to complex devices such as disk array controllers. As one skilled in the art will appreciate, it is not likely that a standard for managing disk array controllers will be created in the future, because unlike host adapters and disk subsystems, each disk array vendor is constantly releasing proprietary features to distinguish itself in the marketplace. It is well known in the art that devices, such as storage systems, can connect directly to the network using, for example, an ethernet or other suitable network adapter. Each device connected directly to the network has an IP address identifying itself on the network. Thus, devices

SUMMARY OF THE INVENTION

According to the invention, a system and method for monitoring and managing devices on a network. The system and method preferably comprises a proxy server connected to the network and a managed device connected to the proxy server. The system farther comprises storage means for storing a device management application program associated with the managed device, and a management station in communication with the managed device via the proxy server and in communication with the storage means. The management station preferably is configured to retrieve the device management application program from the storage means and process the device management application program. As the management station processes the device management application program, the management station is able to monitor and manage the managed device.

In accordance with one preferred embodiment of the present invention, the managed device preferably is connected to the proxy server by a suitable communication connection. The communication connection may comprise peripheral component interconnect (PCI), small computer system interface (SCSI), universal serial bus (USB), fiber channel, firewire and the like.

In accordance with yet another embodiment of the present invention, the managed device preferably includes a controller for controlling the managed device. In accordance with this particular aspect of the present invention, when the management station processes the device management application program, the management station is able to send management commands to the controller via the proxy server. Preferably, the managed device is a storage system.

In accordance with yet another embodiment of the present invention, the proxy server preferably includes a device mapper which locates devices connected to the proxy server and assigns a TCP/IP port to each of the devices. Thus, when the device management application program of the management station sends management commands to the controller of the managed device, the device management application program first sends the management commands to the proxy server, and the device mapper in the proxy server routes the management commands to the managed device.

In accordance with yet another embodiment of the present invention, the device management application program preferably communicates with the proxy server using a first communication protocol, and the proxy server communicates with the managed device using a second communica-

Initially, it can be seen that Weber is solving the similar problem, i.e. different devices from different manufacturers are associated with different proprietary operating/control software and/or compatibility problems, as in the present application, wherein the control software resides on the network components and/or devices.

As illustrated in fig. 2 reproduced herein, **device 204-1 preferably includes control software which uses a management application interface program labeled A**. Similarly, devices 204-2 and 204-3 run control software which use a management interface application program labeled B…e.g. col. 6 L55-67 and fig. 2.

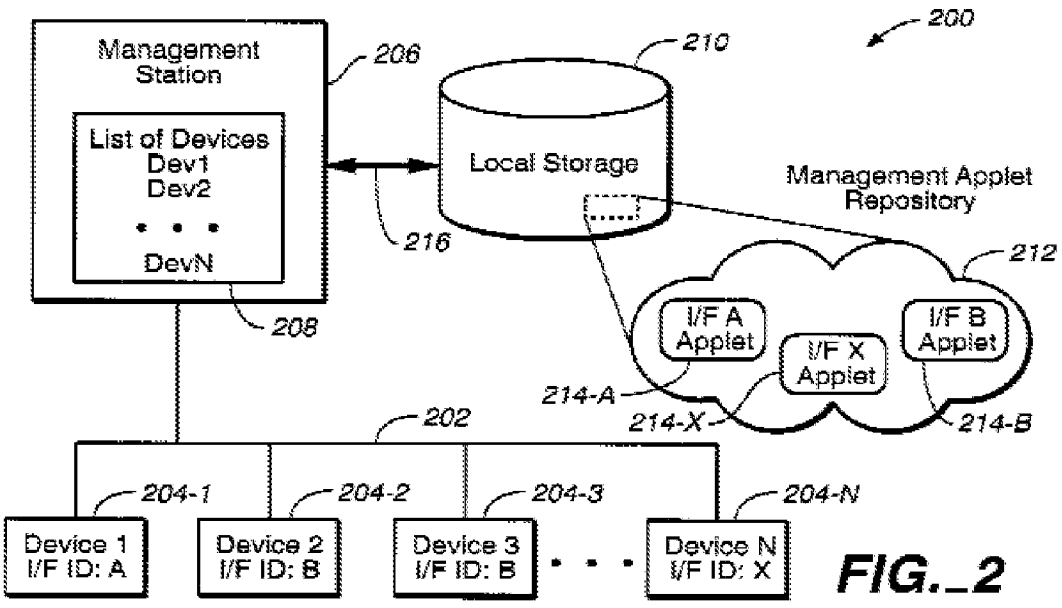

FIG._2

Stated another way, control software and/or application processes are resident on the network components and/or devices, and logically convey which type of management application and/or management interface the network require in order to manage the network device remotely.

As an initial stage, Weber teaches discovering the managed devices on network and presenting them on the management station display, **understanding and maintaining an association between the discovered managed devices and the specific management interface** application program it requires, and providing an interface for invoking the management interface application program for a particular managed device, E.g. col. 9 L10-66.

Weber further discloses:

User Interface

As discussed briefly above, a user preferably interfaces with the management station **510** via a discover-monitor application program **516** and a management interface application program **518** for each of the devices on the network. Preferably, both the discover-monitor application **516** and each of the management interface application programs **518** are Java programs or applets which run in a Java run-time environment.

Discover-monitor Application

Referring now to FIG. 6, a representation of a discover-monitor application screen **600** is illustrated. Preferably, discover monitor application screen **600** includes a management domain window **602**, a detailed information window **604**, a status indicator **606** and a status line **608**.

In accordance with a preferred embodiment of the present invention, management domain window **602** presents a tree structured view of the complete management domain. Lower level nodes **610**, **612** in the tree structure represent actual physical hardware devices such as servers, arrays, and other I/O devices. For example, as illustrated in FIG. **6**, lower level node or server **612-1** includes two storage arrays **610-1** and **610-2** attached thereto. Similarly, lower level node or server **612-2** includes a storage array **610-3** attached thereto. The higher level nodes in the tree represent the location of the hardware devices. For example, in the illustrated embodiment, the management domain is divided into two regions: a central region **618-1** and a southeast region **618-2**. Within central region **618-1**, the domain is further broken into states **616**, for example, Kansas **616-1** and Colorado **616-2**. From there, the domain is further broken down into plant locations **614**, for example, in Colorado **616-2**, the illustrated company has locations in Colorado Springs **614-2** and Fort Collins **614-3**. The management domain shows the servers, storage arrays and other devices which exist on the networks of those locations.

Detailed information window **604** preferably presents the detailed properties for each device in the management domain, based upon the particular node a user selects. Individual device nodes **610**, **612** or a higher level location nodes **614**, **616**, **618** may be selected. When a location is selected, the detailed information window preferably includes an entry for each device in the subtree rooted at the selected location. When a specific device node is selected, detailed information window **604** displays certain device specific attributes of the selected node. In addition, by double-clicking or selecting a specific device node, the device's associated management interface application program is launched.

That is, the management domain enables the user to display information, such as detailed properties for each device as well as specific device attributes, on the at least one application and/or control software residing on the network component through the detailed properties window when the user double clicks the node, i.e. selecting the node, wherein the user interface receives the selection, the device's associated management interface application program is launched, which in particular launches/executes the control software on the node, e.g. col. 7 L15-65.

In column 16 lines 50 to column 17 lines 16, Weber illustrates the launching of the management application program when the user double clicks on one of the storage systems when viewing it in the discover –monitor screen. Next, DMA preferably receives device property information about the selected storage system device from a device property storage area. **Included in the device properties is the storage system's management interface version** (i.e. management application program associated with the device).

That is, the displayed management interface program version in the device properties area is equivalent to displayed information on the at least one determined application processes residing on the network node because of the **understood and maintained association between the discovered managed devices and the specific management interface application program it requires, e**.g. col. 9 L10-66, where the specific management interface application required can only be determined by analyzing and/or understanding the control software of the network node, see fig. 2 above.

For example:

If the version displayed in the device properties area is SNMPv2, then logically, the control software residing on the network node must be SNMP compatible.

Hence, the user interface receives the selection of the one of the displayed information on the least one determined application processes residing on the network node when the user initiates the selection by double clicking the node or object.

b. Weber does not teach launching the selected application process on the selected network component using the determined communication interface from the rules file (Brief, pg. 22: 2nd-4th paragraph).

In response to argument [b], Examiner respectfully disagrees.

In column 16 lines 50 to column 17 lines 16, Weber illustrates the launching of the management application program when the user double clicks on one of the storage systems when viewing it in the discover –monitor screen. Next, DMA preferably receives device property information about the selected storage system device from a device property storage area, wherein the area includes storage system's management interface version, i.e. determining communication interface from the storage (i.e. management application program associated with the device). Next, DMA retrieves from applet repository **residing on the web server or some other location the management interface application program version specified** in the device properties for the selected device, which in particular launches/executes the application and/or control software of the network node, e.g. col. 7 L15-65.

For example:

The specified program version can be in form of SNMPv2. The SNMPv2 indicates **both** the communication interface as SNMP and parameter name as SNMPv2, to use to launch/execute the control software of the network device.

That is, the user interface launches the application process on the selected network component using the selected displayed information by using the determined communication interface, i.e. management application program version from the storage.

As per rules file, as indicated in the prima facie case of obviousness, Weber does not explicitly disclose a rules file having the information such as interface version, etc. However, Weber does disclose using the information from the storage.

Ismael discloses a rules file in form of an html file in a storage device comprising information regarding the network components, e.g. col. 7 L44 to col. 8 L67.

As such, the combination of Weber and Ismael does teach launching the application process, i.e. control software on the network device, using the information such as interface type, version, etc., from the rules file which is stored in the storage device.

On page 21 of the Brief, appellant argues that the cited management interface application is not an application process on the selected network component as claimed.

In response, Examiner agrees that management interface program is not an application process on the network node.

In fact, as set forth above, the user interface determines the management interface to use by using the version information from the storage area in launching the management interface, which launches the application process and/or control software of the network node. In other

words, the management interface program is equivalent to communication interface program and control software is equivalent to application process.

(i) Weber does not use information on the device to invoke an application residing on the network device, but instead uses the information to invoke management application locally to communicate with the remote device (Brief, pg. 23: 1st paragraph).

In response to (i), Examiner disagrees.

As set forth in response to argument [b], the usage of the specified management application program version to invoke the management application program, which invokes/executes the control software and/or application process, shows the usage of communication interface type and parameter name.

For example: The specified program version can be in form of SNMPv2. The SNMPv2 indicates both the communication interface and parameter name, e.g. SNMPv2, to use to launch the control software of the network device.

(ii) Further, the claims require two levels of selection, one of the displayed network component and another of one application process associated with the network component (Brief, pg. 23: 2nd paragraph).

Similarly, Weber discloses two levels of selection, one of the displayed network component, i.e. selecting the node on the display monitor for retrieving detail information and double clicking to launch the management application to

launch the control software and/or application process residing on the network node, e.g. col. 7 L15-65, col. 9 L10-42, col. 13 L10-67, col. 16 L50 to col. 17 L25.

c. Nowhere does Ismael teach or suggest a rules file having, for each of a plurality of network components, a communication interface and parameter that are used to launch the application process on the network component (Brief, pg. 24: 1st paragraph).

In response to argument [c], Examiner respectfully disagrees.

It seems that appellant is addressing the prima facie case of obviousness by attacking references individually.

MPEP 2145 (IV) discloses that one cannot show nonobviousness by attacking references individually where the rejections are based on the combinations of references. In re Keller, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); In re Merck & Co., Inc., 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986).

As set forth in response to argument [b], Weber discloses storing in a storage device for each of a plurality of network components, a communication interface and parameter name that are used to launch the application process on the network component, e.g. the management interface program version indicates the communication interface type and parameter, e.g. col. 7 L15-65, col. 9 L10-42, col. 13 L10-67, col. 16 L50 to col. 17 L25.

For example:

The specified program version can be in form of SNMPv2. The SNMPv2 indicates both the communication interface and parameter name, e.g. SNMPv2, to use to launch the control software of the network device.

However, Weber does not disclose the rules file.

Ismael discloses the rules file in form of html, which is stored in the storage device.

As such, the combination of Weber and Ismael does teaches a rules file having for each of plurality of network components a communication interface and parameter name that are used to launch the application and/or control software on the network device.

d. Nowhere do the cited sections teach or suggest graphical objects representing application processes on network components (Brief, pg. 25 [2]).

In response to argument [d], Examiner respectfully disagrees.

Dependent claim 2 recites:

> The system of claim 1, further comprising:
> **a graphical output device coupled to the interface process for** displaying one or more graphical objects **representing** the application processes on the network components, wherein the interface process is coupled to the graphical output device **for** effecting the display of the graphical objects on the graphical output device.

Weber explicitly discloses a graphical output device coupled to the interface process, e.g. fig. 1 item #112, 114, 116, fig. 4 item #406 with a browser, fig. 5 item #510 with browser or Java Runtime, for displaying the graphical objects representing the application processes on the network components, e.g. fig. 6 item #602 or management domain with storage systems which when double clicked launches the management application and the control software of the

storage system, col. 13 L1 to col. 14 L41. **The objects and/or nodes on fig. 6 are representing the application/control process on the network node.**

e. The cited section do not each or suggest that an application process in a network component is executed in response to selecting an object representing the application process of the network component (Brief, pg. 26 [3.]).

In response to argument [e], Examiner respectfully disagrees.

As set forth in response to argument [a-b], it can be seen that when the user double clicks and/or selects a particular node on the display of fig. 6, which is a graphical display with plurality of **nodes as objects**, the management interface program is launched which executes the control software on the network node in order to manage the network node, e.g. col. 6 L55-67 and col. 7 L15-65.

In other words, the objects on display of fig. 6 are representing the control and/or application processes that are resident on the network components, which can simply be executed by double clicking the objects on the display of fig. 6.

f. The cited section do not teach or suggest upon selecting a network component to access a store to identify application processes residing on that selected network component (Brief, pg. 27 [4.]).

In response to argument [f], Examiner respectfully disagrees.

Dependent claim 6 recites:

The system of claim 5, wherein the interface process accesses the store, upon selection of one graphical object representing one of the network components, **to** identify at least one application process residing on the selected network component represented by the selected object.

In column 16 lines 50 to column 17 lines 16, Weber illustrates the launching of the management application program when the user double clicks on one of the storage systems when viewing it in the discover –monitor screen, i.e. graphical screen. Next, DMA preferably receives device property information about the selected storage system device from a device property storage area, wherein the area includes storage system's management interface version (i.e. management application program associated with the device). Next, DMA retrieves from **applet repository residing on the web server or some other location the management interface application program version specified** in the device properties for the selected device or object, i.e. it identifies and launches/executes the application and/or control software of the network node via the management interface program, e.g. col. 7 L15-65.

Furthermore, "the fact that appellant has recognized another advantage which would flow naturally from following the suggestion of the prior art cannot be the basis for patentability when the differences would otherwise be obvious." Ex parte Obiaya, 227 USPQ 58, 60 (Bd. Pat. App. & Inter. 1985). MPEP 2145 II.

Moreover, the specified version in Weber indicates and/or identifies the compatible and/or proprietary control software resident on the network device.

For example: SNMPv2 will indicate and identify the control and/or application process residing on the network device. It identifies the control software and/or application process as SNMP compatible.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

(12) Conclusion

The components in a SAN are typically acquired from different vendors, and have various application software, i.e. control software, associated therewith (see applicant specification page 174 lines 10-17 and Weber, col. 1 L40-67).

The claimed invention in the present application solves the problem of executing these vendor-specific applications from a single graphical user interface, i.e. a communication interface such as web browser or SNMP application, which effects the launching of the vendor-specific application residing in the device, similar to the process disclosed by Weber (see applicant specification, pg. 174 lines 19-21 and Weber, fig. 1, fig. 4 and col. 4 L9-31).

Information regarding the device-specific management application, i.e. application process, as in Weber is retrieved from a storage which when selected, launches and effects the execution of the control software residing in the network component enabling the administrator to manage the device (col. 7 L15 to col. 8 L15).

Logically, the management application and/or communication interface type such as SNMP provides the information on the control software residing in the device and effects the execution of the control software through the management application or interface, which would then enable administrator to manage the device.

Displaying the management application on the management server is logically identifying the application process at the managed device. Thus, when the management server executes and launches the management application, it actually executes and launches the application/control software at the remote device.

Without the execution of the control software, the management of the vendor-specific devices would be impossible.

As such, the combination of Weber and Ismael discloses and teaches the claimed invention.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

Kamal Divecha
Art Unit 2151
/John Follansbee/
Supervisory Patent Examiner, Art Unit 2151

Conferees:

/John Follansbee/

Supervisory Patent Examiner, Art Unit 2151

/Jeffrey Pwu/

Supervisory Patent Examiner, Art Unit 2146